(6

Generation of k-ary Trees. Extended abstract

(10) Chunglaung ~~C.L.~~ Liu
Department of Computer Science,
University of Illinois,
Urbana, Illinois 61801, USA

(11) 1980

(12) 9

(extended abstract)

1. Introduction

The problem of generating ordered trees was studied rather extensively in the recent literatures [1,2,3,4,5,6,7]. As in all problems concerning the generation of a set of combinatorial objects, there are three major aspects of the problem. After defining a certain linear ordering over the set of combinatorial objects that we are interested in we need to (i) design an algorithm for generating the combinatorial objects one by one, (ii) design an algorithm for determining the rank (the relative position in the linear ordering) of a given combinatorial object, and (iii) design an algorithm for generating the combinatorial object when its rank is given. There, in most cases, is an additional consideration, namely, to choose a suitable way to represent the combinatorial objects. (For example, trees can be represented by sequences of 0's and 1's, sequences of integers, and permutations of integers). Therefore, we are actually concerned with the generation, ranking, and unranking of a chosen representation of the combinatorial objects.

(15)

This work was partially supported by the US Office of Naval Research under the Contract Number N00014-79-C-0775 and by the Institut National de Recherche d'Informatique et d'Automatique, Le Chesnay, France.

176011

Furthemore, a chosen representation might possess a natural linear ordering (for example, the lexicographical ordering of sequences of integers) which might or might not be consistent with the linear ordering over the combinatorial objects that was defined earlier. In this paper, we study ranking and unranking algorithms for k-ary trees when they are represented by sequences of 0's and 1's and sequences of integers from 1 to n where n is the number of internal nodes of a tree.

By a (regular) k-ary tree, we mean an ordered tree in which every internal node has (exactly) almost k sons. We define a linear ordering of the set of k-ary trees as follows :

Definition 1 : Given two k-ary trees T and T', we say that $T < T'$ if (i) T is empty (i.e. T has only a single node) and T' is not empty, or (ii) T is not empty, then for some $1 \leq i \leq k$, $T_j = T'_j$ for $j = 1, 2, \ldots\ldots, i-1$ and $T_i < T'_i$, where $T_1, T_2, \ldots\ldots, T_k, T'_1, T'_2, \ldots\ldots, T'_k$ denote the subtrees of T and T', respectively.

It is well-known that a regular k-ary tree can be represented by a sequence of 0's and 1's, if we label an internal node with a 1 and a leaf with a 0 and traverse the tree in preorder (root - left - right), then the sequence of n(k-1)+1 0's and n 1's so obtained is a unique representation of the tree. Since the last digit in such sequences is always a 0, by convention, we drop the last 0. It can be shown that [6,7] :

Theorem 1 : A sequence of n 1's and (k-1)n 0's represents a k-ary tree if and only if in any prefix of the sequence, the number of 1's is larger than or equal to (k-1) times the number of 0's.

We shall refer to such a representation as the 0-1 representation of k-ary trees. It can be shown that :

Theorem 2 : The lexicographical ordering of the 0-1 representations of k-ary trees is consistent with the linear ordering in Definition 1.

Another way to represent a k-ary tree is by a sequence of positive integers which are the level numbers of the leaves of the tree, read off in the left to right order. By the level of a leaf, we mean the number of internal nodes in the path from the root to the leaf. It can be shown that [3,4] :

Theorem 3 : A sequence of $(k-1)n+1$ integers $a_1 a_2 \ldots\ldots a_{k(n-1)+1}$ where $1 \le a_i \le n$ for $i = 1, 2, \ldots\ldots, k(n-1)+1$ represent a k-ary tree if the sequence is reduced to 0 by a sequence of left reductions. (By a left reduction of a sequence we mean to replace the left-most k consecutive identifical integers $q\ q \ldots\ldots q$ by the integer $q-1$).

We shall refer to such a representation as the level-number representation of k-ary trees. It can be shown that :

Theorem 4 : The lexicographical ordering of the level-number representations of k-ary trees is consistent with the linear ordering in Definition 1.

Figure 1 shows an example.

3. A Reduction Scheme

We show in this section a reduction scheme which is [illegible] time [illegible]. Given a regular k-ary tree with n internal nodes, we remove first the left most k-1 leaves of the tree and then reconstruct a regular k-ary tree with n-1 internal nodes as in the follows : Let y be the left most son of x in a k-ary tree. If y has less than k sons, remove y and let the sons of y become sons of x as shown in Figure 2(a). If y has more than k sons, let the right most excessive sons become sons of x as shown in Figure 2(b). (k is equal to 3 in the examples in Figure 2(a) and (b)). Now, suppose the left most k-1 leaves have been removed from a k-ary tree. Consider the sequence of internal nodes $x_1, x_2, \ldots\ldots, x_i$ where x_1 is the right brother of the first leaf removed and x_i is the father of the k-1st leaf removed. Applying the reduction step to x_i and then x_{i-1}, and then $x_{i-2}, \ldots\ldots$, we shall obtain a regular k-ary tree with n-1 internal nodes. (We give here only an intuitive description of the reduction step. Although it is not totally obvious that the reduction step will indeed always yield a regular k-ary tree, these will all become very clear in next section.) Let T be a regular k-ary tree. We shall use $R(T)$ to denote the regular k-ary tree obtained by the reduction procedure described above. Figure 3 shows an example. It can be shown that :

Theorem 5 : Let $\mathcal{T}$ be the set of all k-ary trees with n internal nodes with the level numbers of the first k-1 leaves being $a_1, a_2, \ldots\ldots, a_{k-1}$. Let $R(\mathcal{T})$ denote the set of k-ary trees obtained by the reduction procedure described above. Then $R(\mathcal{T})$ is the set of all k-ary trees with n-1 internal nodes and with the level number of the first leaf being $a_{k-1}-1$ or larger. Furthemore, for T and T' in $\mathcal{T}$, if $T < T'$ then $R(T) < R(T')$.

3. Ranking and Unranking Algorithm

We show now a ranking and an unranking algorithm for k-ary trees when they are represented by sequences of 0's and 1's. Let $\underline{a}$ denote the 0-1 representation of a tree T. Let $\underline{\hat{a}}$ denote the 0-1 representation of the tree $R(T)$. It can be shown that :

Theorem 6 : $\underline{\hat{a}}$ is obtained from $\underline{a}$ by deleting the first k-1 0's and the right most 1 in front of the k-1st 0.

Let index($\underline{a}$) denote the number of k-ary tree of n internal nodes whose 0-1 representation is larger than $\underline{a}$ in the lexicographical ordering of 0-1 sequences. Let $A(a_1, a_2, \ldots\ldots, a_i)$ $(i \leq k-1)$ denote the number of k-ary trees with the first i level numbers being $a_1, a_2, \ldots\ldots, a_i$. It can be shown that :

Theorem 7 : Let $\underline{a}$ denote the 0-1 representation of a k-ary tree with its first k-1 level numbers being $a_1, a_2, \ldots\ldots, a_{k-1}$.

Then

$$\begin{aligned}\text{index}(\underline{a}) &= A(a_1+1) + A(a_1,\ a_2+1) + A(a_1,\ a_2,\ a_3+1)\\ &\quad +\ldots\ldots+ A(a_1,\ a_2,\ldots\ldots,\ a_{k-1},\ a_k+1)\\ &\quad + \text{index}(\underline{\hat{a}})\\ &= N(n+1,\ a_1+1,\ k-1) + \sum_{i=2}^{k-1} N(n,\ a_i+1,\ i-2)\\ &\quad + \text{index}(\underline{\hat{a}})\end{aligned}$$

where $$N(n,\ b,\ i) = \frac{kb-b-i}{kn-b-i}\binom{kn-b-i}{n-b}.$$

Theorem 7 is the basis of a ranking algorithm, the details of which should be quite obvious.

The unranking algorithm is the reverse of the ranking algorithm which can be described as follows : suppose we are given index(T).

(1) a_1 is determined to be the smallest integer j such that index(T) < $N(n+1,\ j+1,\ k-1)$

(2) a_2 is determined to be the smallest integer j such that index(T) - $N(n+1,\ a_i+1,\ k-1) < T(n,\ j,\ 0)$

(3) $a_3,\ldots\ldots,\ a_{k-1}$ is determined recursively to be the smallest integer j such that

$$\text{index}(T) - N(n+1,\ a_1+1,\ k-1) - \sum_{i=2}^{p-1} N(n,\ a_i+1,\ i-2)$$

$$< N(n,\ j,\ p-2)$$

for $p = 3,\ldots\ldots,\ k-1$

It should be noted that the ranking and unranking algorithm presented here are natural extensions of that discovered by Zaks [6].

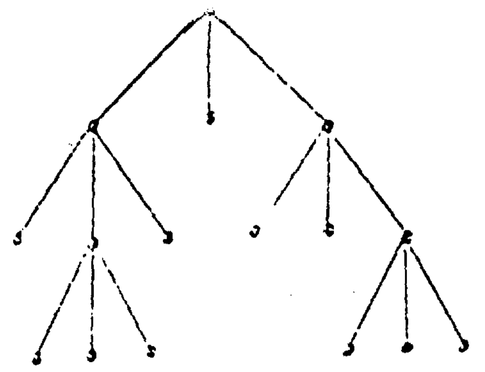
It should be evident how the ranking and unranking algorithm presented above can be modified when the level-numbers representations of k-ary trees are employed. Indeed, let $\underline{l}$ denote th level-number representation of T and $\dot{\underline{l}}$ denote the level-number representation of $R(T)$. Theorem 7 can be restated as :

Theorem 8 : let $\underline{l}$ denote the level-number representation of a k-ary tree with its first k-1 leaves being $a_1, a_2, \ldots\ldots, a_{k-1}$. Then

$$\begin{aligned} \text{index}(\underline{l}) &= N(n+1,\ a_1+1,\ k-1) \\ &+ \sum_{i=2}^{k-1} N(n,\ a_i+1,\ i-2) \\ &+ \text{index}(\dot{\underline{l}}) \end{aligned}$$

The only difference is how to obtain $\dot{\underline{l}}$ from $\underline{l}$. This can be done by carrying out a sequence of left reductions to identify the subtrees of internal nodes which are moved up in the reduction scheme. The level numbers of the leaves of such subtrees [illegible] l. We leave the details to a forthcoming report.

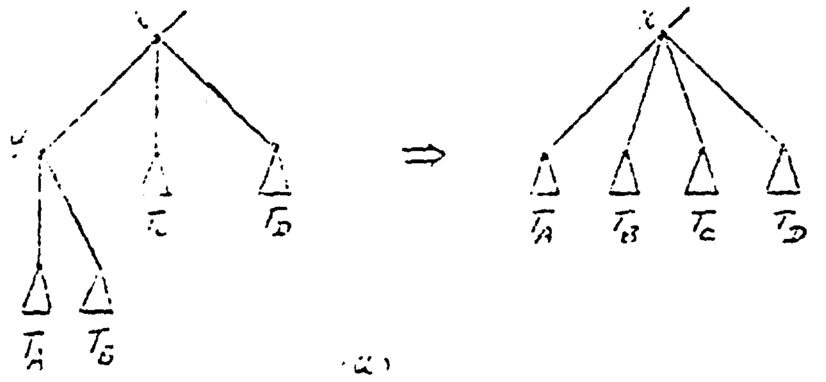
Our results can also be applied to the case in which a regular binary tree is represented by a permutation as defined in Trojanowski [5]. Again, we leave the details to a forthcoming report.



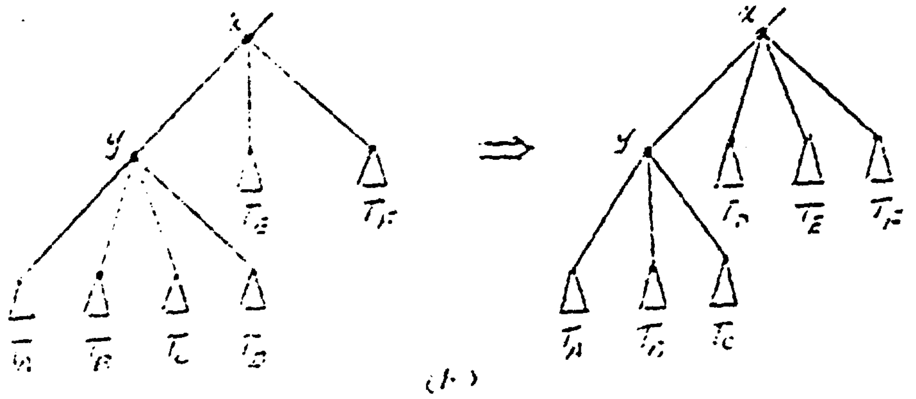
0-1 representation : 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 $\underline{0}$

Level number representation 2 3 3 3 2 1 2 2 3 3 3

Figure 1



(a)



(b)

Figure 2

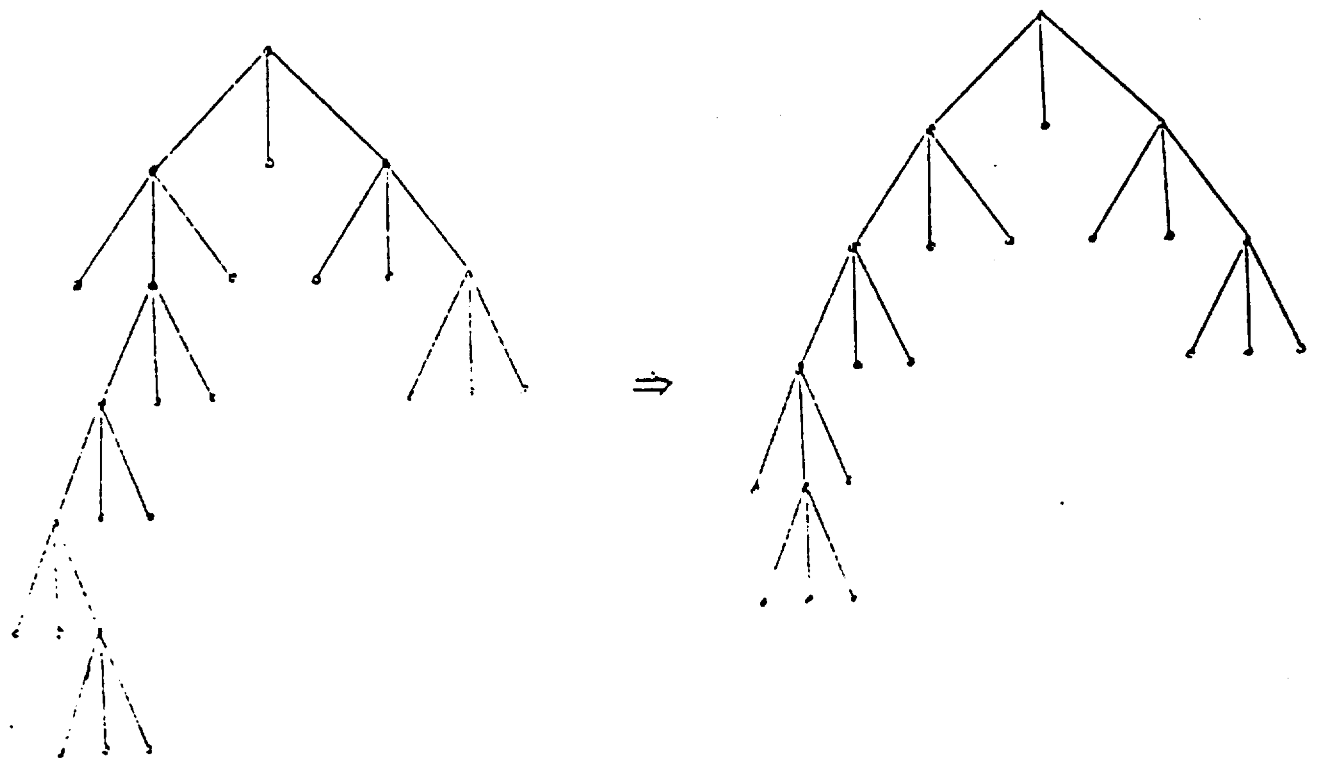


Figure 3

References

[1] Knott, G.D., A numbering system for binary trees, Comm. ACM, 20 (1977), 113-115.

[2] Roten, D., and Y.L. Varol, Generating of binary trees from ballot sequences, J. ACM, 25 (1978), 396-404.

[3] Ruskey, F., Generating t-ary trees lexicographically, SIAM J. Comput., 7 (1978), 424-439.

[4] Ruskey, F., and T.C. Hu, Generating binary trees lexicographically, SIAM J. Comput., 6 (1977), 754-758.

[5] Trojanowski, A.E., Ranking and listing algorithms for k-ary trees, SIAM J Comput., 7 (1978), 492-509.

[6] Zaks, S., Lexicographic generation of ordered trees, Th. Comput. Sci., 10 (1980), 63-82.

[7] Zaks, S., and D. Richards, Generating trees and other combinatorial objects lexicographically, SIAM J. Comput., 8 (1979), 73-81.